ABSTRACT

An extension of object-oriented programming languages
enables the use of replaceable classes and virtual
constructors, thereby allowing existing code that creates

5   objects of a base class to be reusable for creating instances
of a new class derived from the base class. A base class
intended to be replaceable later is declared to be
replaceable. In the case of a compiled language, the
programming code defining and using the base class is compiled

10 into a reusable module. Later, a new class derived from the
base class is identified as a replacement for the base class
and is used in new programming code that uses the existing
reusable module. The replacement relationship between the old
base class and the new class is registered. During execution

15 of the program, when the old code of the reusable module
indicates the creation of an object of the base class, the
registered replacement information is referenced. If the old
class is to be replaced with the new class, the creation
information for the new class is passed instead so that an

20 object of the new class is created. The class replacement may
be implemented with compiled and interpreted programming
languages. When a compiled language is used, the class
replacement may be implemented with new compiler-supported
features that emit data for class replacement into the .obj

25 files, but may also be implemented without requiring changes
to existing compilers.